SENA

PROGRAMA DE FORMACIÓN

ANÁLISIS EXPLORATORIO DE DATOS EN PYTHON

INSTRUCTOR

LUZ NEIRA VARON PEÑA

AA3-IMPLEMENTAR LAS HERRAMIENTAS Y LIBRERÍAS NECESARIAS PARA EL ANÁLISIS DE LOS DATOS.

APRENDIZ

ENMANUEL A. DUARTE CÁCERES

COLOMBIA

2025

Introducción

En el presente informe se presenta la evidencia de implementación de las herramientas y librerías necesarias para el análisis de datos en Python.

Actividad correspondiente a la semana 3 del programa de formación *Análisis exploratorio de datos en Python*.

A lo largo del documento se detallan los pasos para la correcta manipulación, lectura, ordenamiento y limpieza de datos, correspondiente al manejo de la infraestructura para el correcto análisis de datos, así mismo se realizan diferentes cálculos estadísticos y se visualizan para la correcta interpretación.

Finalmente, el documento muestra las conclusiones de la actividad.

Caso de estudio

El dataset utilizado para el desarrollo de la actividad se encuentra en el archivo CSV titulado "DatosSeguros". A continuación, se muestra una tabla correspondiente al tipo de variable y las columnas que lo conforman.

Tabla 1. Variables del Dataset.

Variables Categóricas	Variables Numéricas
Sexo (object)	Edad (int64)
Fumador (object)	Imc (float64)
Region (object)	Valor_seguro (float64)
	Hijos (int64)

Preguntas objetivo:

1. ¿Las personas fumadoras representan el mayor potencial para ventas de seguros?
2. ¿Cuál es la edad poblacional en la que se debe concentrar la estrategia comercial teniendo en cuenta el mayor ingreso?
3. ¿Cuál es la región con mayor potencial de venta de seguros teniendo en cuenta los ingresos?

Para la correcta manipulación debemos hacer uso de diferentes librerías, las cuales nos ayudaran a reducir el tiempo de ejecución y nos brindan las herramientas necesarias para el correcto procesamiento y análisis de datos.

Tabla 2. Librerías utilizadas.

Librería	Descripción
Pandas	Usada para la manipulación y visualización de grandes volúmenes de datos Comando: `Import pandas as pd`
Matplotlib	Permite generar muy fácilmente diversos tipos de gráficos. Comando: `Import matplotlib.pyplot as plt`
Seaborn	Permite la gestión de gráficos más atractivos de Matplotlib, gráficos informativos y estadísticos. Comando: `Import seaborn as sns`

Creación del Dataframe

Para iniciar debemos crear un dataframe que lea los datos en nuestro CSV

```
df = pd.read_csv('DatosSeguros.csv')
```

y visualizamos la información en el dataframe

```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1349 entries, 0 to 1348
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   edad          1349 non-null   int64
 1   sexo          1349 non-null   object
 2   imc           1347 non-null   float64
 3   hijos         1349 non-null   int64
 4   fumador       1347 non-null   object
 5   region        1349 non-null   object
 6   valor_seguro  1349 non-null   float64
```

Se puede visualizar en la información de los datos que el resultado para las columnas imc y fumador es de 1347, mientras que para las demás columnas es de 1349. Esto indica que existen valores vacíos para esas dos variables.procedemos a limpiar los datos.

```
df = df.dropna()
```

y con el siguiente comando eliminamos los duplicados

```
df = df.drop_duplicates()
df.info()
<class 'pandas.core.frame.DataFrame'>
Index: 1333 entries, 0 to 1347
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   edad          1333 non-null   int64
 1   sexo          1333 non-null   object
 2   imc           1333 non-null   float64
 3   hijos         1333 non-null   int64
 4   fumador       1333 non-null   object
 5   region        1333 non-null   object
 6   valor_seguro  1333 non-null   float64
```

Ahora visualizamos la información en la siguiente tabla:

Tabla 3. Tabulación de los datos.

Total, registros	1349
Registros sin valores nulos	1345
Total, nulos	4
Total, duplicados	12
Registro sin nulos y duplicados	1333
Variables categóricas	Sexo, fumador, región
Variables numéricas	Edad, imc, hijos, valor_seguro

Ordenamiento y agrupación de datos

Menor a mayor

```
edad = df.sort_values('edad')
```

Mayor a menor

```
edad = df.sort_values('edad',ascending = False)
```

Definimos los rangos

```
rangos = [17,28,38,48,58,68]
```

Y asignamos un nombre o etiqueta a cada uno

```
nombrerangos = ['A','B','C','D','E']
```

Ahora con este nuevo agrupamiento podemos crear una nueva variable

```
df ['Rango_Edad'] = pd.cut(df['edad'],rangos,labels = nombrerangos)
```

con el siguiente comando se puede visualizar los primeros registros del data frame,

```
df.head()
```

Tabla 4. Visualización primeros registros.

	edad	sexo	imc	hijos	fumador	region	valor_seguro	Rango_Edad
0	19	F	27.900	0	yes	Caribe	16884.92	A
3	18	M	33.770	1	no	Cundinamarca	1725.552	A
4	28	M	33.000	3	no	Cundinamarca	4449.462	A
5	33	M	22.705	0	no	Antioquia	21984.47	B

6	32	M	28.880	0	no	Antioquia	3866.855	B

Análisis estadístico

A continuación, se utiliza la función describe(), el cuál permite calcular las medidas de tendencia central y dispersión, aplicando algunos métodos estadísticos como la media, mediana, desviación estándar y cuartiles.

```
df.describe()
```

Tabla 5. Medidas de tendencia central y dispersión.

	edad	imc	hijos	valor_seguro
count	1333.000000	1333.000000	1333.000000	1333.000000
mean	39.195049	30.652097	1.092273	13261.908454
std	14.052008	6.097609	1.205484	12093.507648
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.220000	0.000000	4738.268200
50%	39.000000	30.360000	1.000000	9377.904700
75%	51.000000	34.675000	2.000000	16657.717450
max	64.000000	53.130000	5.000000	63770.428010

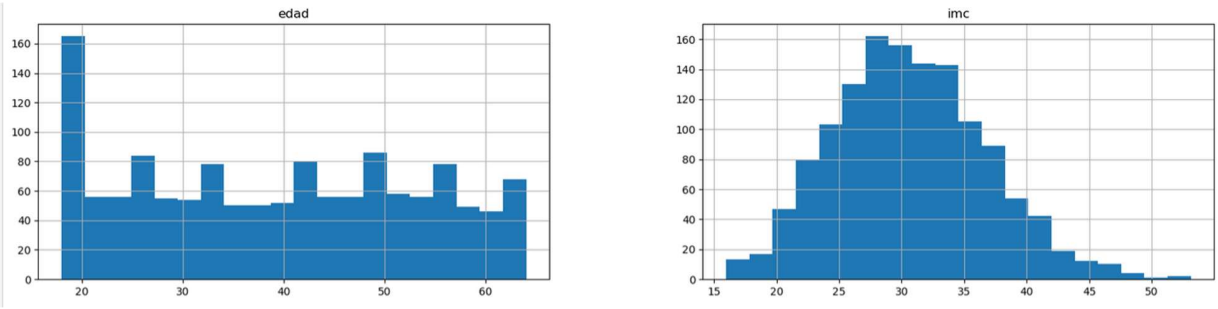
En la tabla 5 se puede ver como la media y la mediana tienen valores muy cercanos, el único atributo que presenta una marcada diferencia es el valor_seguro.

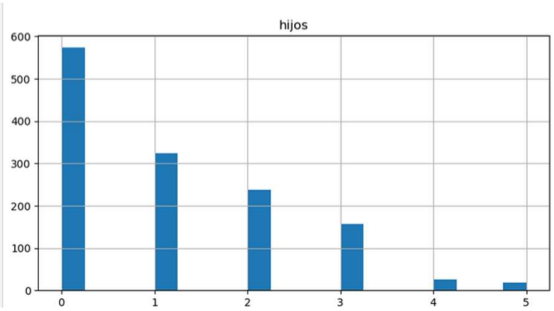
Gráficos

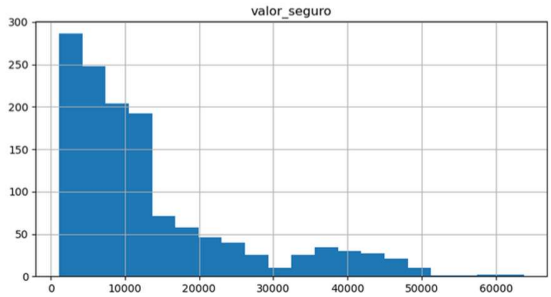
Para una mejor comprensión de los datos se utilizan los gráficos, en Python podemos utilizar histogramas de frecuencia, gráficos de barras, gráfico de torta, gráficos de cajas y bigote.

Histograma de frecuencia

```
df.hist(bins=20, figsize=(20,10))
```

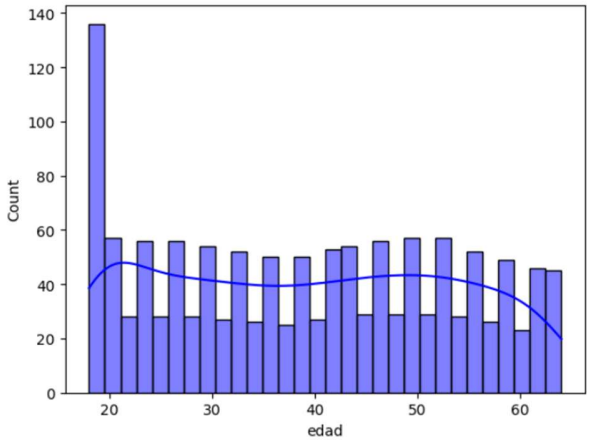




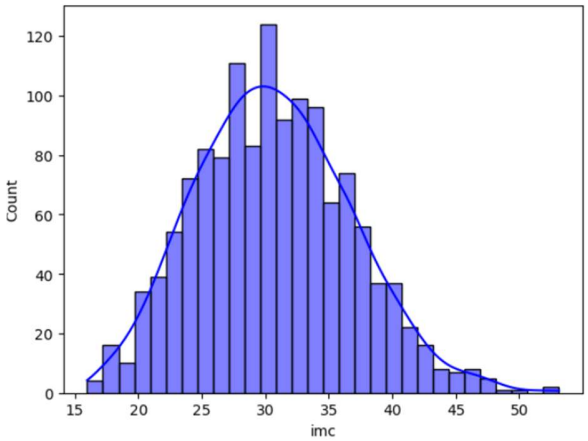


Para mejorar estéticamente los gráficos se utiliza el siguiente comando, con el cuál podremos visualizar una línea suavizada mostrando la distribución de los datos.

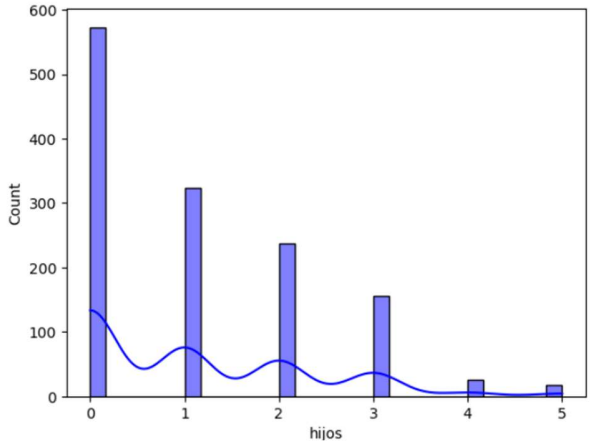
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.histplot(df.edad, color =”b”, bins = 30, kde = True)
plt.show()
```




```
sns.histplot(df.imc, color =”b”, bins = 30, kde = True)
plt.show()
```




```
sns.histplot(df.hijos, color =”b”, bins = 30, kde = True)
plt.show()
```




```
sns.histplot(df.valor_seguro, color =”b”, bins = 30, kde = True)
plt.show()
```

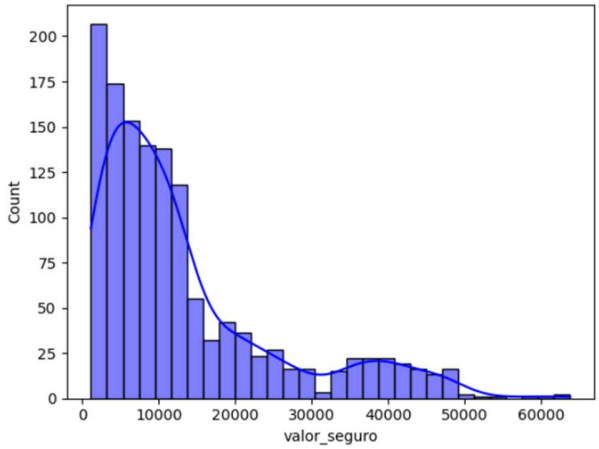
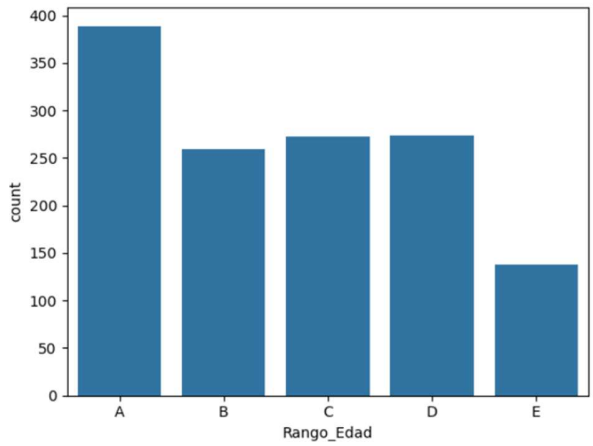


Gráfico de barras

```
plt. Figure(figsize=(10,7))
sns.countplot(x= df.Rango_Edad)
plt.show()
```

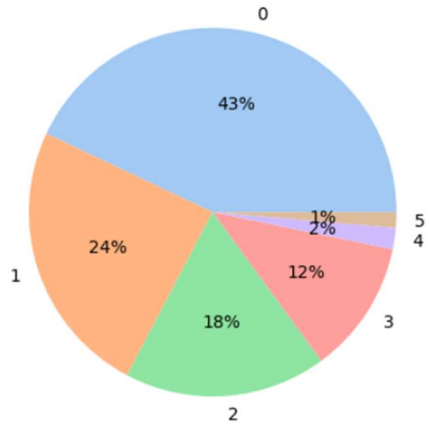


A partir de los histogramas de frecuencia y grafico de barras podemos decir que hay una mayor agrupación de registros en el Rango de 18 a 28 años, así mismo, el imc tiende a tener una distribución normal, donde los datos están entre los 25 y 35, y la mayoría no tiene hijos, seguidos por aquellos que tienen 1 solo hijo; existiendo una mayor concentración en el valor de seguro hasta los 10000.

Gráficos de Torta

Distribución por cantidad de hijos

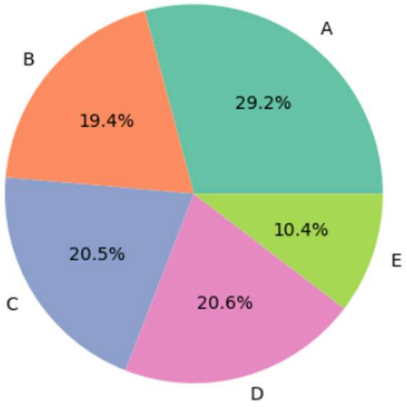
```
total_hijos = df['hijos'].groupby(df['hijos']).count()
etiquetas = total_hijos.index
colors = sns.color_palette('pastel')[0:6]
plt.pie(total_hijos, labels = etiquetas, colors = colors,
autopct='%.0f%%')
plt.show()
```



La mayoría tiene entre 1 y 3 hijos, muy pocos son los registros (3 %) que tienen 4 o más hijos, por lo que las familias que tienes un seguro según los datos son pequeñas o medianas

Distribución por rango de edades

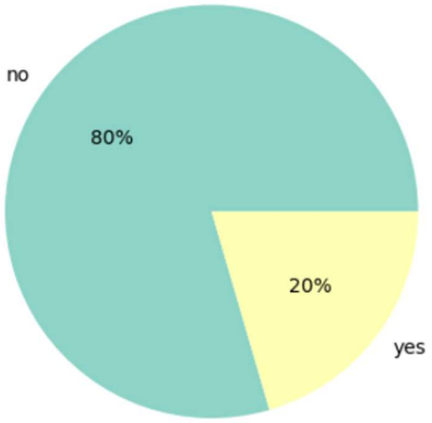
```
total_rango_edad = df['Rango_Edad'].groupby(df['Rango_Edad'],
observed = True).count()
labels = total_rango_edad.index
colors = sns.color_palette('Set2')[0:5]
plt.pie(total_rango_edad, labels = labels, colors = colors,
autopct='%.1f%%')
plt.show()
```



La mayoría de registros se encuentran agrupados entre los rangos A y C, lo cual deja solo un 31% para los mayores de 48 años (grupos D y E)

Distribución de fumadores

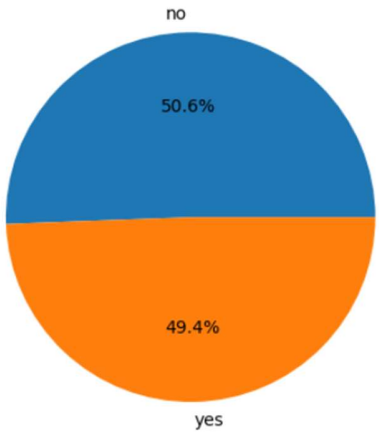
```
total_fumadores = df['fumador'].groupby(df['fumador']).count()
labels = total_fumadores.index
colors = sns.color_palette('Set3')[0:2]
plt.pie(total_fumadores, labels = labels, colors = colors,
autopct='%.0f%%')
plt.show()
```



El gráfico es contundente con la información del dataset podemos decir que la base de fumadores representa solo el 20% de los registros.

Distribución de compra de seguro por categoría fumadores

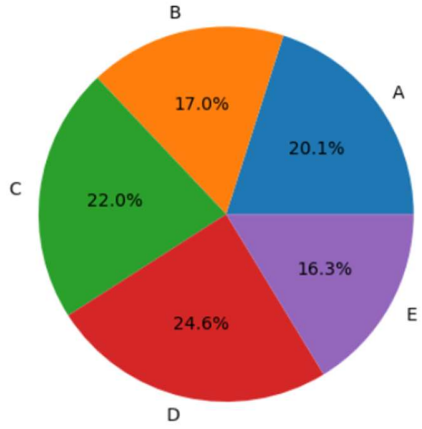
```
valor_total_por_fumador =
df.groupby('fumador')['valor_seguro'].sum()
plt.pie(valor_total_por_fumador.values, labels =
valor_total_por_fumador.index, autopct ='%.1f%%')
plt.show()
```



Los valores entre no fumadores y fumadores están muy parejos, aunque sabemos que hay menos no fumadores, ellos dan casi la misma cantidad de ingresos que el grupo de no fumadores

Distribución de valores porcentualmente pagados al seguro de acuerdo con el rango de edad

```
valor_total_por_rango_edad = df.groupby('Rango_Edad', observed=True)['valor_seguro'].sum()
plt.pie(valor_total_por_rango_edad.values, labels = valor_total_por_rango_edad.index, autopct ='%.1f%%')
plt.show()
```

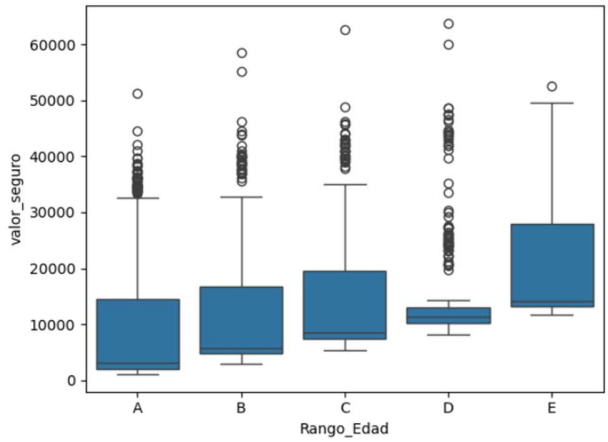


Porcentualmente no hay una gran diferencia entre los ingresos de cada Rango, pero podemos ver que el Rango E representa un menor ingreso correspondiente al total, mientras que la mayor concentración está en el Rango D.

Gráficos de Caja y Bigotes

Relación entre el valor de seguro y el rango de edades

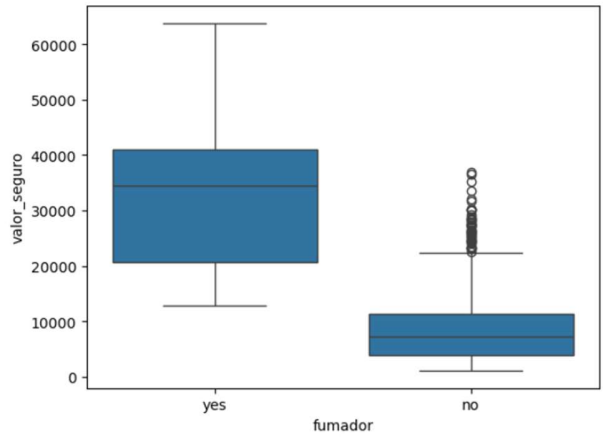
```
Redad_valor = sns.boxplot(x=df[”Rango_Edad”], y=df[”valor_seguro”])
```



Se observa que la mayor mediana se encuentra en el rango E, a pesar de ser los que dan menos ingresos (ver gráfica anterior), también hay que enfatizar que el rango con mayores valores atípicos es el rango D.

Relación del valor de seguro con fumadores y no fumadores

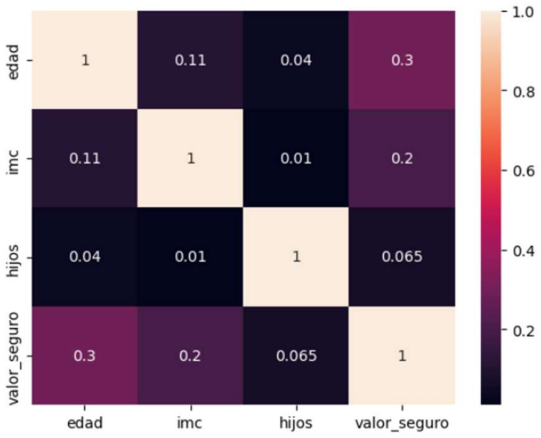
```
fumador_valor = sns.boxplot(x=df[”fumador”],
y=df[”valor_seguro”])
```



Del gráfico podemos concluir que los fumadores son los que representan una mayor fuente de ingresos al tener una mediana y limites superiores e inferiores superiores a los no fumadorse.

Correlación de variables

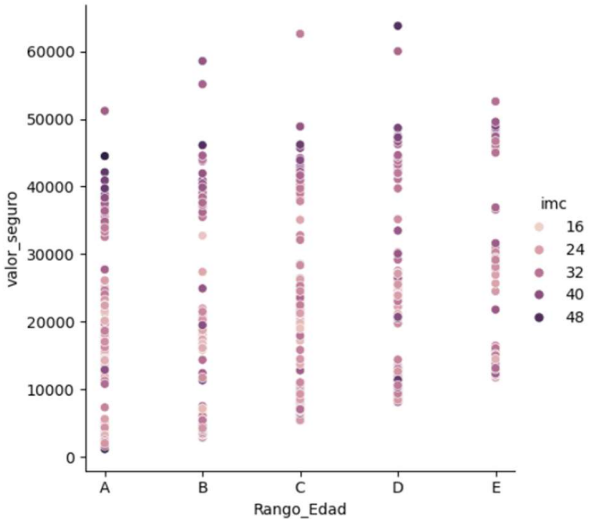
```
correlacion = df.corr(numeric_only=True)
sns.heatmap(correlacion,xticklabels=correlacion.columns,yticklab
els=correlacion.columns,annot=True)
```



Relaciones multivariable

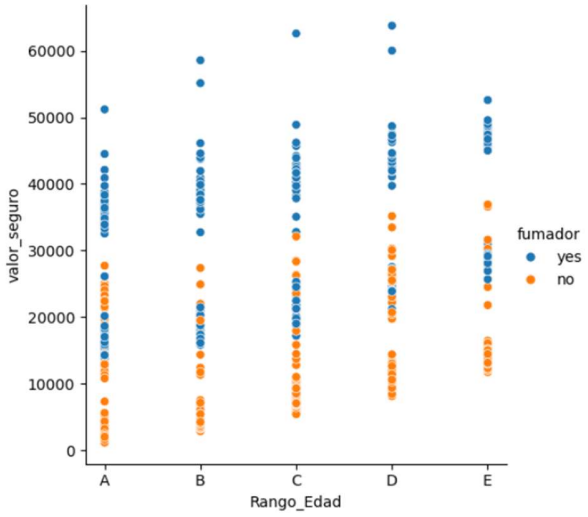
A continuación, podemos observar gráficamente la relación entre el valor de seguro, con el rango de edad y el imc.

```
sns.relplot(x='Rango_Edad' ,y='valor_seguro',hue='imc',data=df)
```



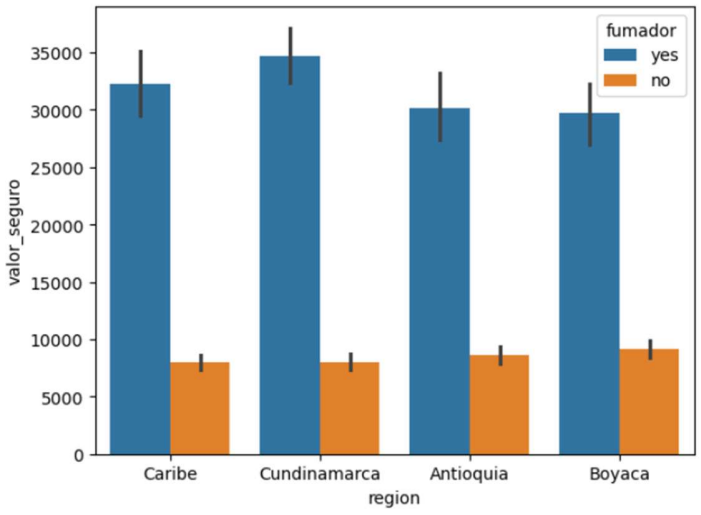
Ahora haremos lo mismo con la relación entre el valor de seguro, con el rango de edad y el fumador.

```
sns.relplot(x='Rango_Edad',
y='valor_seguro',hue='fumador',data=df)
```



Finalmente haremos el análisis multivariado con los valores de seguro, fumadores y región.

```
sns.barplot(data=df,x='region',y='valor_seguro',hue='fumador')
```



Conclusiones

Partiendo de las preguntas planteadas y los resultados obtenidos en el análisis de un solo dato y multivariado, se pueden generar las siguientes conclusiones:

- El 80 % de los registros analizados no son fumadores.
- Cerca del 45 % de los clientes no tienen hijos.
- Aproximadamente, el 70 % de los clientes tiene al menos un hijo.
- El rango de edad para el rango A representa cerca del 30 % del total de registros.
- Los valores pagados al seguro por los clientes identificados como fumadores equivalen al 50 % del total.
- A pesar de que los clientes no fumadores equivalen al 80 %, solo representan el 50% del valor total del seguro.
- La región no es determinante en el valor del seguro.
- Lo más relevante en los valores pagados al seguro se dio en personas que son fumadores